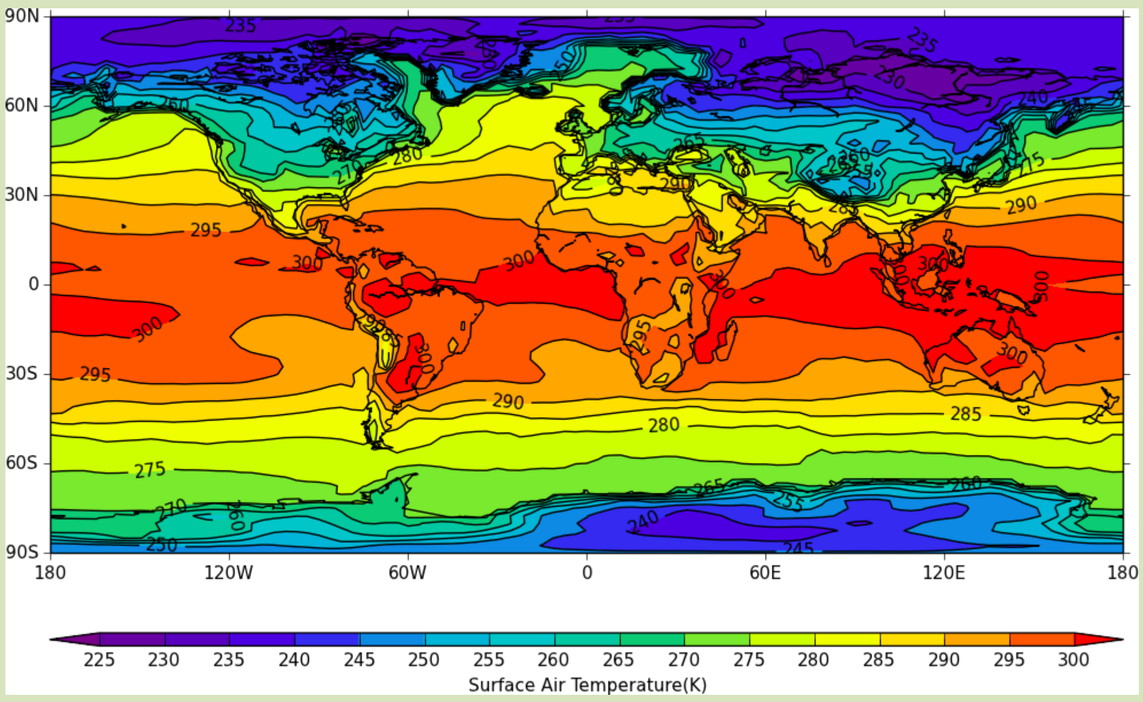
New analysis and visualisation tools

cf-python and cfplot

Andy Heaps*, David Hassell* and Charles Roberts*
*NCAS-CMS

Computational Modelling Services
NATIONAL CENTRE FOR ATMOSPHERIC SCIENCE
NATURAL ENVIRONMENT RESEARCH COUNCIL

Introduction




```
>>> import cf, cfplot
>>> f = cf.read('/opt/graphics/cfplot_data/tas_A1.nc')
>>> cfplot.con(f)
```

cf-python

A python package which creates, reads, writes and manipulates data and its metadata. It is fully compliant with the CF metadata conventions [1] and can read data stored in CF-netCDF, Met Office (UK) PP format files and Met Office (UK) fields files.

It implements Large Amounts of Massive Arrays (LAMA) functionality, which allows multiple fields larger than the available memory to be manipulated without any extra work from the user.

cfplot

A python package that uses the metadata in a CF field to create an appropriate contour or vector plot. The package allows easy user control of colour tables and plot positioning.

Metadata-aware software

Both **cf-python** and **cfplot** are metadata-aware, meaning that they make full use of a field's metadata (such which axes it has) to simplify the analyses. In the simple example shown above, the field read from the file has global coverage with latitude and longitude axes, a fact which didn't need to be known nor specified by the user.

Manipulate

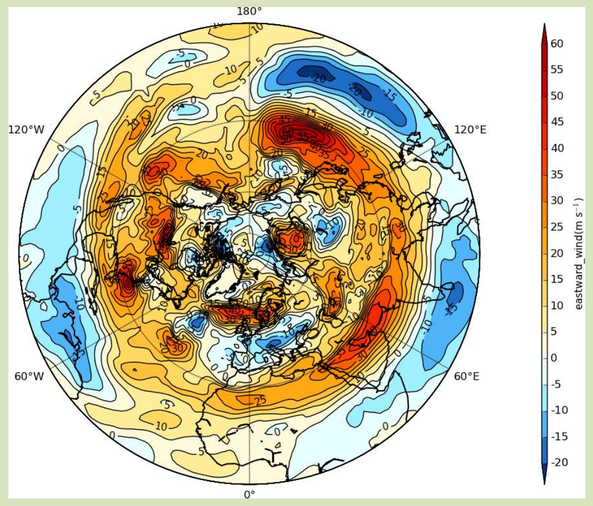
Visualise

- Use the **import** function to import the **cf** and **cfplot** packages.
- Use the **cf.read** function to read a file and select the air temperature field from it
- Use the **print** function to print information about the field.

```
>>> import cf, cfplot
>>> U = cf.read('/opt/graphics/cfplot_data/ggap.nc', select='eastward_wind')
>>> print U
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(23), latitude(160), longitude(320)) m s**-1
Axes           : time(1) = [1964-01-21 00:00:00]
               : pressure(23) = [1000.0, ..., 1.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
```

- Use the **subspace** method to select the air temperature at 500 hPa
- Print information about the new field

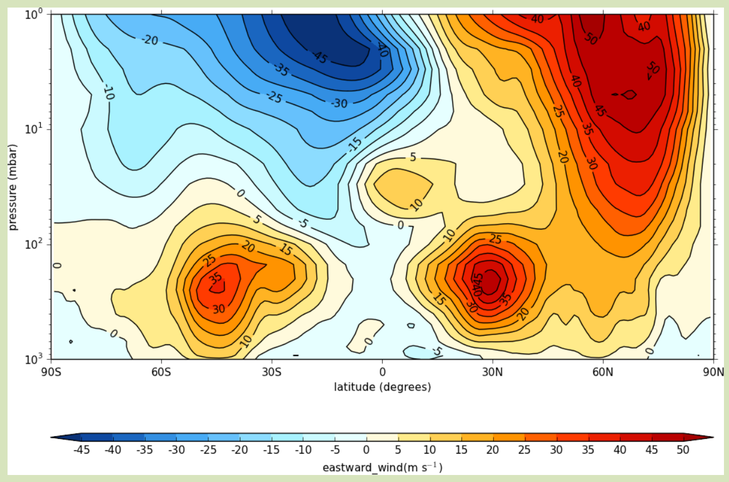
```
>>> U_500 = f.subspace(pressure=500)
>>> print U_500
eastward_wind field summary
---------------------------
Data           : eastward_wind(time(1), pressure(1), latitude(160), longitude(320)) m s**-1
Axes           : time(1) = [1964-01-21 00:00:00]
               : pressure(1) = [500.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(320) = [0.0, ..., 358.875] degrees_east
```




```
>>> cfplot.mapset(proj='npstere')
>>> cfplot.con(U_500)
```

- Use the **cf.collapse** function to calculate the zonal mean air temperature for each latitude and at each pressure level.
- Print information about the new field

```
>>> zonal_mean = cf.collapse(U, 'mean', axes='X', weights='X')
>>> print zonal_mean
----------------------------
Data           : eastward_wind(time(1), pressure(23), latitude(160), longitude(1)) m s**-1
Cell methods   : longitude: mean
Axes           : time(1) = [1964-01-21 00:00:00]
               : pressure(23) = [1000.0, ..., 1.0] mbar
               : latitude(160) = [89.1415176392, ..., -89.1415176392] degrees_north
               : longitude(1) = [179.4375] degrees_east
```




```
>>> cfplot.con(zonal_mean)
```

Documentation

Google **cf-python** and **cfplot** for further examples and full documentation.

Future Developments

- Performance optimisations (parallelisation)
- Fully conservative field regridding
- cfview – gui interface to cf-python and cfplot